

INTERACCIÓN FÍSICA DENTRO DE UN MODELO DE REALIDAD AUMENTADA, PROYECTO ITACA

Mt. Univ. Juan Manuel Corso Sarmiento
Arquitecto
Junio 2011

TECNOLOGÍA

La creación del código unipersonal NIV (número de identificación virtual) para un único usuario se realiza a través de un código de respuesta inmediata “QRcode”, que se puede leer en cualquier dispositivo móvil que tenga una cámara, una pantalla y acceso a internet a través del código abierto “OpenCV”, que permite alimentar en tiempo real una Base de datos online. Elejemplo desarrollado se baso en las aplicaciones de Antares, en unity3D, figura 1.

¿Qué se logra con éste código único?

1. En el museo físico se ofrecen los contenidos y los servicios de forma personalizada a cada perfil de usuario según sus características e intereses: edad, género, idioma, formación, preferencias, tiempo, etc haciendo posible que la información sea más accesible. La adopción de esta tecnología también permite la interacción en forma de realidad virtual y/o aumentada.
2. En el museo web presenta dos ventajas: la primera, cuando el usuario finaliza la visita al museo, le permite continuar su experiencia de forma remota (desde su casa, ordenador u otro lugar) utilizando su código NIV como acceso directo a su perfil. Segunda, si el usuario ha ampliado la información de su perfil, le permite preparar su visita previa al museo, facilitando y mejorando su experiencia en su recorrido futuro al museo real. Su página personal le permitirá aportar contenidos y compartirlos para que otros usuarios puedan valorarlos.
3. En la entidad del Museo Marítimo, este código único, al gestionarse a través de internet, permite alimentar en tiempo real una base de datos de ambos museos (físico y el web), por lo que ésta puede evaluar y mejorar continuamente las preferencias y necesidades de los usuarios que lo visitan.

La propuesta se basa en la implementación de un sistema que permita adaptarse fácilmente a cualquier entorno, así como una rápida modificación, basándose en tecnologías pre-existentes y de amplio uso. El NIV requiere únicamente de una cámara web para ir a la base de datos y regresar con el vínculo a los contenidos específicos, se puede visualizar en cualquier dispositivo informático que cuente con una pantalla, una cámara y conexión a internet.

Más que un solo programa pre-empaquetado destinado a grupos con discapacidades específicas, esta tecnología ofrece la posibilidad de permitir a cada visitante obtener lo contenidos adecuados a su perfil (ej. niño, experto, sordo, chino, etc). Al ser una tecnología basada en la web, los visitantes del museo podrán utilizar el NIV para acceder a contenidos relacionados (como los exámenes de exposiciones, críticas de arte, o la información período histórico) en tiempo real mientras recorren el museo físico.

Cada visitante a través del perfil de su NIV puede tener distintas experiencias en el museo que reflejen sus preferencias personales, intereses y estados de ánimo, así como su adaptación a necesidades especiales.

Para su funcionamiento el sistema requiere de la implantación de las siguientes

QRCODE (Quick Response Code/ Código de Respuesta Rápida)

Un QRcode es un código de barras de dos dimensiones, que permite almacenar distintas informaciones y que se puede imprimir. Es un “hardlink” es decir, un hipervínculo entre el mundo físico y el mundo web.

La capacidad de datos del código QR es:

Numérico Máx. 7.089 caracteres

Alfanumérico Máx. 4.296 caracteres

Binario Máx. 2.953 bytes

Al ser un código imprimible, se puede aplicar en cualquier superficie del mundo físico, y puede ser reconocido por cualquier equipo que cuente con una cámara web no menor a 640x480 pixeles, una pantalla y conexión a internet. Con el programa adecuado puede generarse un código para cada usuario y vincular a ese código una respuesta según se requiera o información que contenga la base de datos.

Figura 1: Codificación y decodificación de códigos QR, aplicación Antares sobre la plataforma Unity3D



Fuente: Propia

- OPENCV

El reconocimiento de estos códigos se realiza a través del estándar OpenCV (Open Source Computer Vision). Es una biblioteca de funciones de programación dirigida principalmente al análisis de imágenes digitales. Entre los ejemplos más significativos de la aplicación de esta biblioteca podemos encontrar: Human-Computer Interaction (HCI), objetos de identificación,

segmentación y reconocimiento de la cara y gestos; propuestas de seguimiento de imágenes, Robótica Móvil, etc.

Es una aplicación de código abierto, lo que la convierte en una tecnología económica de implementar, tampoco requiere de ningún nivel de formación o previa instrucción para su utilización, por lo que su mayor ventaja es que es de fácil manejo para cualquier tipo de usuarios.

TÉCNICAS ACCESIBLES PARA EL CONTROL DE MODELOS VIRTUALES, PROYECTO ITACA

Objetivo del proyecto:

El objetivo del proyecto ITACA se centra en ofrecer soluciones tecnológicas para crear entornos accesibles para todos, que permitan a los usuarios disfrutar su visita a los entornos patrimoniales de forma segura y adaptada a las características de cada persona. Este proyecto busca el desarrollo de nuevas soluciones tecnológicamente avanzadas, así como la adaptación al ámbito de los edificios patrimoniales de la tecnología domótica y sistemas de control de entorno que han sido aplicadas hasta la fecha a soluciones de accesibilidad desarrolladas para entornos residenciales. La adaptación de los entornos incorporando nuevas soluciones tecnológicas permite situar al usuario en el centro de un entorno de inteligencia ambiental. Los edificios de patrimonio incorporan importantes peculiaridades debido a los tipos de soluciones constructivas y a las fuertes restricciones estéticas, la adaptación a estos entornos de soluciones existentes implica la superación de grandes retos derivados de estas peculiaridades.

FACEAPI RECONOCIMIENTO DE ROSTROS PARA VIDEOJUEGOS

El reconocimiento de objetos (especialmente de rostros) es un campo que esta obteniendo cada vez más trascendencia, como por ejemplo el trabajo de Johny Lee, con el seguimiento a través del wiimote, utilizándola como una cámara de infrarrojos, obteniendo un resultado bastante impresionantes, otras opciones son utilizar cámaras especiales, como la ps3eye, que permite manejar el puntero del mouse como si fuera un mouse, permitiendo una variedad ilimitada de aplicaciones, como las detalladas en los siguientes enlaces:

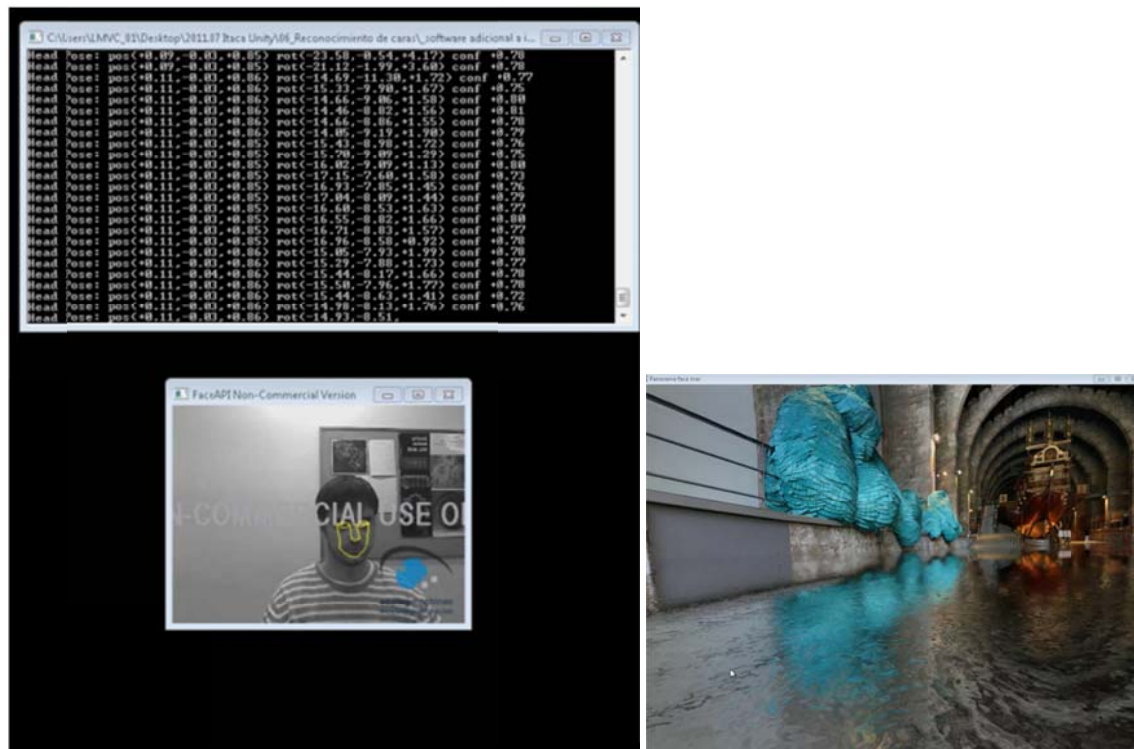
http://en.wikipedia.org/wiki/PlayStation_Eye

<http://forums.eagle.ru/showthread.php?t=56837&page=15>

Otra opción más económica y muy práctica es el reconocimiento de rostros a partir de la óptica de una cámara web, en la cual no necesitamos ni luces invisibles en nuestras cabezas ni cámaras que solo sirven para esa utilidad (figura 2). Para ello existen bibliotecas de software que permiten dicho reconocimiento como faceAPI.

Como ejemplo de ello esta el siguiente video, en el que estoy navegando en un panorama con los movimientos de mi cabeza. Siendo interesante ya que se navega por un panorama que tiene información 3D complementaria, como el agua o viñetas.

Figura 2: x, y, z, pitch, yaw, roll, mediante la detección de rostros y la interacción en un panorama



Fuente: Propia

Guía rápida para faceAPI desde Unity 3D:

El primer paso es registrarse en faceAPI para descargar la versión no comercial del programa en el siguiente enlace: <http://www.faceapi.com/Download/downloads.php> o en <http://www.seeingmachines.com/product/faceapi/downloads/>

Para la transmisión de la información utilizaremos una aplicación llamada faceapistreamer0.95, la cual podemos descargar aquí: <http://code.google.com/p/6dofstreamer/downloads/list>

A continuación, abra Socket.exe para verificar que todo funciona correctamente. Si usted no ha instalado la licencia no comercial, sin embargo, que reciba no puede encontrar errores .dll

Este streamer funciona mediante el envío de datos UDP a través de la red a la unidad. Envía los datos como una cadena en este orden separado por comas (x, y, z, pitch, yaw, roll). Ahora todo lo que tiene que hacer es recibir los datos UDP en el interior de la Unidad y lo revisará en las variables adecuadas.

Eso es todo, iniciar el Socket.exe para transmitir los datos y pulse el botón Reproducir en la Unidad.

Nosotros estamos utilizando la versión no comercial de FaceAPI como un ejemplo, pero otros 6 grados de libertad trackers están disponibles. El código abierto mejorado Human Computer Interface (EHCI) de proyecto con un streamer similar se puede utilizar también (aunque se necesita para implementar el streamer)

La descripción más detallada desde la creación de la escena la encontramos en el siguiente enlace:

<http://visionblaster.com/cms/images/stories/pdfs/streamerdocumentation0.95.pdf>
<http://code.google.com/p/ehci/>.

Otros ejemplos de tecnicas similares las vemos con el desarrollo comercial de la Kinetic, en los cuales los videojuegos están pasando a otro nivel. Como ven en los ejemplo de abajo.

<http://www.youtube.com/watch?v=t7xwnsB2IMQ>
<http://www.youtube.com/watch?v=uu5Se...eature=related>
<http://www.youtube.com/watch?v=LTcQEhppeZI>

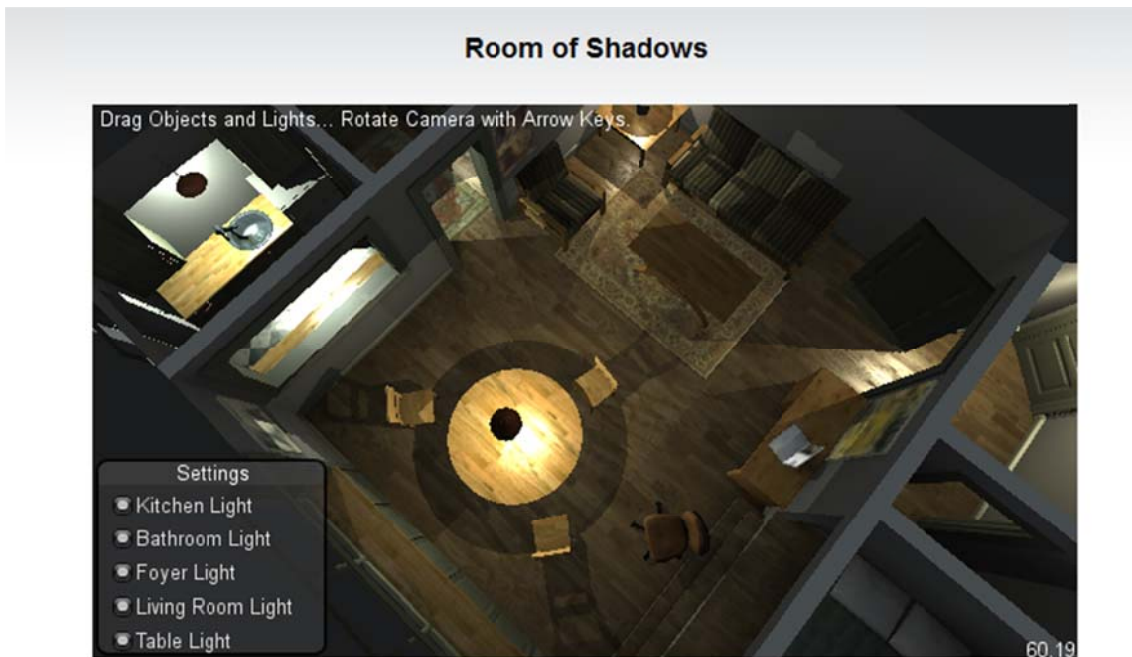
ARRASTRAR OBJETOS CON EL MOUSE UNITY3D

En términos de interacción con plataformas de videojuegos siempre existe la misma problemática ¿Cuál es la mejor forma para interactuar con el contenido?, sobre la cual no hay una respuesta única, ya que depende de la idea central de la aplicación.

En la mayoría de casos estas interacciones pretenden ser intuitivas, eliminando la barrera que genera el desconocimiento de la aplicación. En el siguiente ejemplo mostraremos como manipular libremente objetos de una escena en Unity3D.

En primer lugar podemos ver un ejemplo de interacción en los demos de la página de Unity3D en el siguiente enlace: <http://unity3d.com/gallery/live-demos/shadows>. El cual es un poco más complicado que solamente utilizar Rigidbody, ya que se resaltan los objetos de rojo y se juega con las luces y sombras (figura 4).

Figura 4: Demo de Unity3D



Fuente: <http://unity3d.com/gallery/live-demos/shadows>

Para realizar esta escena se utiliza el script DragRigidbody.js (este proceso utiliza Raycast, Rigidbody y SpringJoint para lograr el arrastrar y soltar), siguiendo los siguientes pasos:

1. Crear una escena que cuente con un suelo o terreno
2. Agregar un GameObject como por ejemplo un cubo
3. Asignar Rigidbody a los que desea ser capaz de arrastrar (Component-Physics-Rigidbody)
4. Agregar First Person Controller desde los Prefabs de Standard Assets
5. Finalmente, arrastre el DragRigidbody.js en el objeto First Person Controller de la escena.

Si estudiamos la demo de Room of Shadow de Unity3D podemos exportar el comportamiento de iluminación y selección de objetos de dicha escena, lo cual vemos en el siguiente video, como un ejemplo que prepare con una idea para un proyecto de estudio (Figura 5).

Figura 5: Ejemplo de interacción en el museo marítimo de Barcelona



Fuente: Propia